IN-THE-FIELD ADAPTATION OF A LARGE VOCABULARY AUTOMATIC SPEECH RECOGNIZER (ASR)

BACKGROUND OF THE INVENTION

5   **Technical Field**

The present invention relates generally to automatic speech recognition (ASR) and, more particularly, to the process of tuning a speech recognition engine to improve its accuracy.

10 **Description of the Related Art**

Speech recognition is an imperfect art. Achieving high accuracy is difficult because multiple variables typically exist including, e.g., differences in microphones, speech accents, and speaker abilities. When

15 Automatic Speech Recognition is performed over a telephone network, the task is even more difficult, owing to the noise and bandwidth limitations imposed on the speech signal.

It is known in the prior art to tune a speech

20 recognition engine to increase the engine's level of accuracy. In the simplest example, speaker adaptation, such tuning is effected in a completely supervised manner, with the user of the system being prompted to read given text over a period of time. During this process, the

25 speech recognizer is adapted to the user's voice. Examples

of this approach are found in many commercial products,

such as Dragon Dictate. These techniques generally require

several minute sessions between the user and the system,

and they are therefore inappropriate to telephone-based

5   ASR, where most interactions last only a few utterances and

the user identity usually cannot be saved for future

sessions.

For larger, speaker-independent systems, tuning the

recognizer to individual speakers is not practical or

10   desirable. The goal of tuning such systems is to arrive at

generally applicable models and algorithms. Nor is it

possible in these systems to conduct any supervised

sessions with the user population. In such cases, ASR

providers tune their algorithms using human intervention.

15   In particular, after the recognizer is deployed, a large

quantity of speech data is collected. Human listeners then

transcribe this speech data. Transcription requires

careful and skilled listening to each utterance in the

database, as well as excellent typing ability. Using the

20   speech data and the human-provided transcriptions, the ASR

provider then tunes the recognition engine as necessary and

re-deploys the application. This type of tuning is not

economical, and it is often not rapid enough to be useful

in deploying large vocabulary ASR systems. Indeed, as the

- 2 -

size of the vocabulary increases, such "supervised" tuning

techniques become more inefficient and can fail to bring

the system up to desired level of accuracy in a practical

amount of development time and expense. The present

5   invention addresses this problem.

BRIEF SUMMARY OF THE INVENTION

It is a primary object of the present invention to improve the global recognition accuracy at an ASR installation in an automated manner.

5   It is another primary object of this invention to bring an ASR algorithm up to an acceptable accuracy level in the field and without direct supervision.

Another important object of this invention is to provide a novel method of tuning a large vocabulary speech

10   recognition engine in an economical and reliable manner. These and other objects of the invention are provided by tuning the ASR engine in a completely automated manner. In particular, the speech recognition results from the recognizer, which typically is imperfect when first

15   deployed, are used to train or tune one or more aspects of the ASR model and algorithm. This in-the-field adaptation depends on the recognizer to transcribe the database, as opposed to a human listener. It then relies on automated tuning techniques to actually modify the algorithm, as

20   opposed to expert human algorithm developers.

Thus, according to the present invention, the imperfect output of the recognizer itself is preferably the only information used to supervise the transcription of

live input speech data. This data is then analyzed by one

or more adaptation algorithms, and the recognizer is re-

tuned as necessary to increase its recognition accuracy.

The invention provides a unique method for adapting a

5   speaker-independent ASR algorithm to a given application or

environment in an unsupervised manner in the field. To

tune the ASR, one or more site adaptation algorithms are

used, either singularly or in combination. These are

preferably speaker-independent algorithms that are intended

10   to learn such application-specific features as channel

characteristics, dialects, pronunciation idiosyncrasies,

and speaking style. The adaptation may be performed on the

acoustic model (e.g., Hidden Markov Models), the language

model (e.g., Word Bigram Statistics), the pronunciation

15   model (e.g., phonetic dictionary), or combinations thereof.

The foregoing has outlined some of the more pertinent

objects and features of the present invention. These

objects should be construed to be merely illustrative of

some of the more prominent features and applications of the

20   invention. Many other beneficial results can be attained

by applying the disclosed invention in a different manner

or modifying the invention as will be described.

Accordingly, other objects and a fuller understanding of

the invention may be had by referring to the following

- 5 -

Detailed Description of the Preferred Embodiment.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present

invention and the advantages thereof, reference should be

made to the following Detailed Description taken in

5   connection with the accompanying drawing in which:

Figure 1 is a flowchart illustrating the in-the-field

adaptation technique in accordance with the present

invention;

Figure 2 is a graph illustrating a relative error

10   reduction as a function of an amount of training data

according to a Hidden Markov Modeling (HMM) adaptation

algorithm;

Figures 3A and **3B** are tables illustrating the results

of testing a Word Bigram Statistics adaptation algorithm;

15   **Figures 4A** and **4B** are tables illustrating the results

of testing a phonetic dictionary adaptation;

Figure 5 is a graph illustrating the results of

combining two adaptation algorithms, the HMM adaptation,

and the Word Bigram Statistics adaptation.

20

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 illustrates a flowchart of the in-the-field adaptation methodology of the present invention. The invention assumes that an ASR engine has been deployed at a

5   user installation and that the provider desires to enhance the recognition accuracy of the engine, preferably without human intervention. In one preferred implementation, the invention is used with an ASR engine having a large vocabulary, although this is not a requirement. Thus, for

10   example, the ASR application is a directory assistance database, which is comprised of a large number (e.g., several thousand) vocabulary items.

The routine begins at step 20 by deploying the ASR engine, for example, at a customer installation in the

15   field. It is assumed that the ASR engine as initially deployed is imperfect and must be tuned to increase its accuracy level. At step 22, a test is run to determine whether the ASR engine has been set for automatic adaptation according to the invention. This may be

20   achieved, for example, by extending the engine's application programming interface (API) to include a flag that may be set (e.g., by the application developer or the user) to begin the adaptation process. The flag may be set

- 8 -

remotely after the engine has been deployed. If the engine has been set for automatic adaptation, the routine continues in step 24. At this step, a test is made to determine whether a given data-collection period (e.g., a

5   24 hour period) has elapsed. If not, the routine continues at step 26 to store the recognition results, along with the associated speech waveform samples. In particular, during this step, the recognition results obtained by processing a live input are saved to a disk file. As will be seen, the

10   recognition results may include the actual results (i.e., the hypothesized spoken utterances) generated by the ASR engine, together with information such as confidence levels, n_best hypotheses, and other data which might be used as input to the adaptation algorithms in step 30.

15   In a variant of step 26, the system may be configured to save a "downstream" version of the speech data (e.g., cepstral coefficients), instead of the "raw" digitized speech waveform samples. This is another advantage of the present invention. In particular, because there is no

20   requirement for humans to listen to the speech data, significant data-reduction may be obtained by storing only the form of the speech data that is required for executing the adaptation algorithms. This advantage can result in reduction in costs for computer equipment, including CPUs,

- 9 -

IC memory, and hard disks.

Step 26 cycles until the result of the test at step 24 is positive. At this point, the routine continues at step 28 to retrieve the information saved during the time

5 period. At step 30, an adaptation algorithm (or a plurality of adaptation algorithms) is executed against the information to increase the accuracy of the engine. As will be seen, this algorithm may be based on an acoustic model (e.g., Hidden Markov Modeling), a language model

10 (e.g., Word Bigram Statistics), a pronunciation model (e.g., Phonetic Transcription), or some combination of these different model types. At step 32, the so-tuned recognition engine is then re-installed in the application, presumably with better accuracy and more efficient use of

15 computing resources than the original engine.

This completes the basic processing. Of course, the entire basic process may be repeated, so that the recognition continually improves, and also so that the models are able to adapt to changes in the application

20 conditions that might affect recognition performance.

Thus, according to the present invention, the recognition engine itself performs transcriptions of the live input data and then utilizes such information, along with some form of the original speech signal, as the input

- 10 -

to an adaptation algorithm. The ASR engine is thus tuned in an unsupervised manner to a given application or operating environment in the field, and without human intervention.

5   While batch processing of the collected recognition results is preferred, one of ordinary skill in the art will appreciate that, if sufficient computational and memory resources exist, the adaptation routine can be exercised against the recognition data on-the-fly, namely, as live

10   input data is being recognized.

The adaptation algorithms are preferably speaker-independent, and they are selected to enable the ASR engine to learn such application-specific features as channel characteristics, dialects, pronunciation idiosyncrasies,

15   and speaking style, among others. One type of algorithm is a Hidden Markov Model adaptation, in which the mean-vectors and covariance matrices of the acoustic model are updated according to the acoustic input and corresponding speech states, hypothesized by the un-tuned recognizer. Another

20   type of algorithm is a Word Bigram Statistics adaptation, which is a language model that updates lattice probabilities according to presumed word sequences. Another representative algorithm type is a phonetic transcription adaptation, which is a pronunciation model

- 11 -

that updates a lexicon with alternative transcriptions

generated, for example, from a text-to-phoneme engine and

pruned according to the new data.

Finally, it is well-known that many current ASR search

5 algorithms are parametric. Examples of such algorithms are

Viterbi beam searches, stack decoders, and A-star searches.

In such algorithms, there are many parameters that could

potentially be tuned according to the above method. For

example, the beam-width and the pruning-threshold could be

10 adjusted in a Viterbi beam-search algorithm so that the

search performs more optimally in a given environment.

Also, in a multiple pass search algorithm where the scores

from each pass over the utterance are combined to give a

final score, the method for combining such scores is

15 usually parametric and thus could be tuned according to

this method. The present invention described a general

technique for adapting the parameters in such ASR search

algorithms.

One of ordinary skill in the art will recognize that

20 the above-described approaches (which are merely exemplary)

are potentially orthogonal. Thus, an additive benefit may

be realized according to the invention by combining two or

more of these methods in a single algorithm. Of course,

these methods are not meant to be taken as an exhaustive

list of approaches to in-the-field adaptation according to
the invention.

The following experimental data illustrates that in-
the-field adaptation is useful for improving the field

5   performance of a recognition algorithm.

EXPERIMENTAL

HMM Model Adaptation

In HMM adaptation, the newly captured data is
processed through a traditional Baum-Welch embedded re-

10   estimation algorithm, with only the mean vectors allowed to
be updated. (An alternative implementation, probably
requiring more adaptation data, would also update the
covariance vectors for each state.) As is central to the
unsupervised adaptation method disclosed herein, the best

15   recognition hypothesis for each utterance was used as the
presumed orthographic transcription. Then, the normal
Baum-Welch re-estimation algorithm was applied, allowing
only the mean vectors of each Gaussian model to be updated.
As with the other adaptation approaches described below,

20   several variations on this basic scheme were explored.
First the amount of adaptation data was varied, from about
40 minutes of speech, up to four hours. Second, several
confidence thresholds were applied, including a confidence

- 13 -

gap, wherein only utterances within the two confidence

levels were included in the adaptation process. **Figure 2**

illustrates the results from this experimentation.

As can be seen, the HMM adaptation behaves extremely

5   nicely. The more training data that is used to adapt, the

more relative error reduction is achieved. The improvement

is virtually identical for the development and evaluation

test sets, showing that no over-training has taken place.

A baseline test was also run on a database that was

10   substantially similar in character to the original training

data used to develop the un-tuned recognizer. This test

showed only a small decrease in accuracy, suggesting that

the method is fairly robust and that it will not impact

recognition performance should there be a sudden change in

15   the application characteristics. Also, there was no

significant difference in CPU performance between the

baseline and adapted HMM tests.

Word Bigram Statistics Adaptation

For these experiments, the ASR engine's first-pass

20   Viterbi search graph was biased with word bigram data

extracted from subsets of the recognition results on given

development input data. The result transcriptions (i.e.,

recognition hypotheses) were randomized and then various

sized portions, starting from the top-choice down, were taken to accumulate word pair frequencies. A variation on this experiment imposed a score threshold on the recognition results as the sub-setting mechanism. These

5   counts were converted into probabilities, and these probabilities were used to bias the Viterbi search in favor of the most likely word sequences. The main results of these experiments are summarized in the table illustrated in **Figures 3A-3B**.

10   The main conclusion from these experiments is that recognition results obtained in the field can be used to improve recognition accuracy simply by accumulating the results and recompiling the Viterbi search graph loaded by the recognizer. It is also seen that, as with the HMM

15   adaptation, using the bigram-weighted search graph does not increase CPU consumption.

An extension to the adaptation of word bigram statistics is to adapt word trigram statistics. Indeed, with sufficient adaptation data, this technique may be

20   extended to any of the well-known statistical approaches to language modeling. Moreover, the same basic technique could be applied to deterministic language models, such as BNF-driven context-free grammars.

Phonetic Transcription Adaptation

Another experiment involved adapting phonetic transcriptions in the ASR dictionary. This was accomplished using the following general procedure:

5   1)   A statistical text-to-phoneme algorithm was used to develop a letter-to-phoneme decision-tree, derived from a large U.S. English training lexicon. This decision-tree creates a one-to-many mapping of orthographic spellings to phonetic transcription alternatives, thus over-generating

10   phonetic transcriptions for a given vocabulary item.

2)   This decision-tree was then applied to the test dictionary, producing a probability-ranked list of alternative phonetic transcriptions for each lexical item in the vocabulary.

15   3)   These alternative transcriptions were first pruned to a reasonable number per lexical item, e.g., 20.

4)   Then, these ranked lists were filtered again, by removing phonotactically improbable entries.

5)   Next, the key adaptation step was run. The newly

20   expanded lexicon is used in a forced phonetic alignment on the adaptation data. As with the other techniques, the recognition hypotheses were assumed to be the correct transcription for each utterance. This step therefore uses

the adaptation data to select the most appropriate

transcriptions from the expanded lexicon. Transcriptions

that were infrequently used or not used at all in the

forced alignment were filtered out of the expanded lexicon,

5   reducing the average number of transcriptions from about 18

to about 2, which is typical for large-vocabulary ASR

dictionaries.

6)   Then, the transcriptions selected by this forced

alignment procedure were used to supplement the standard

10   phonetic transcription lexicon.

7)   New tests were then run with this expanded (i.e.

adapted) lexicon.

The tables illustrated in **Figures 4A-4B** show the

experimental results for the Names task.

15   For the development test, the expanded lexicon

obtained a 16.1% relative improvement over the standard

lexicon. This same pattern of results held for the

evaluation test set, but the improvement was not as large.

It may also be desirable to begin to perform

20   adaptation data filtering based on recognition confidence,

as was done in the other in-the-field adaptation

experiments. Further, it may be desired to implement

variations on generating the text-to-phoneme decision-tree,

including tailoring it to specific lexicon types, and, in the context of a names task, using different decision-trees for different linguistic groupings. Or other methods could be employed for generating alternative phonetic

5   transcriptions, including phonological-based approaches.

Combined Approaches

As described earlier, each of the above approaches is fairly orthogonal. Thus, two or more of these adaptation methods may be combined to produce an additive benefit. A

10   simple combination of the HMM and Bigram adaptation generated the results illustrated in **Figure 5**. Combining these adaptation algorithms had a somewhat less than additive effect on the development test data, and a somewhat more than additive effect on the evaluation test

15   data. The combined improvement in both cases approached 18%.

Conclusions

The experimental data demonstrates the plausibility of all three model adaptation methods. Generalizing, these

20   approaches showed a recognition error reduction of nearly 20%, using a combination of two of these approaches, on a real-world test database.

The unsupervised adaptation paradigm of the present

- 18 -

invention has many advantages over the prior art.

Foremost, it does not require human intervention to

supervise the transcriptions of the live input data.

Moreover, adaptation may be effected even on limited data.

5   The results illustrated above reflect that limited amounts

of data can be sufficient to demonstrate improvements.

Further, screening the recognition hypotheses with

confidence thresholds can be effective in improving

adaptation performance. These adaptation methods may be

10 readily implemented to improve real-world ASR engine

accuracy at specific installations.

Many variants are within the scope of the present

invention. Thus, for example, the representation module

(i.e., the basic signal processing and feature extraction)

15 of the ASR algorithm may be adapted to enhance recognition

performance. In addition, more sophisticated approaches to

filtering the adaptation data may be implemented. Further,

"regional" confidence measures – wherein confidences are

attached to individual recognition units (e.g, words or

20 phonemes) – may be developed so more of the data can be

used to adapt the models. In this approach, it is not

necessary to exclude an entire utterance from the

adaptation process because it has a low confidence.

Instead, some of the data from the utterance may be

salvaged and used as input to the model tuning.

The inventive technique may be used within or as an adjunct to a known recognition engine. As is well known, the recognition engine receives a spoken input and may

5   generate multiple recognition hypotheses. This is a known function that is available from several prior art systems (namely, recognition systems, applications and the like) including, without limitation, the Vpro™ speech recognition engine, the VR Speech Recognition engine, and

10   the SpeechWave™ line of speech recognition products, all currently developed and marketed by VCSI. In general, any speech recognition engine that employs acoustic models, language models, pronunciation models, and or a parametric search algorithm may be configured to use the approach

15   described herein.

The algorithms in accordance with the invention preferably comprise software, and thus one of the preferred implementations of the invention is as a set of instructions (program code) in a code module resident in

20   the random access memory of a general purpose computer. Until required by the computer, the set of instructions may be stored in another computer memory, e.g., in a hard disk drive or in a removable memory such as an optical disk (for

eventual use in a CD ROM) or a floppy disk (for eventual

use in a floppy disk drive), or downloaded via the Internet

or some other computer network. In addition, although the

various methods described are conveniently implemented in a

5   computer selectively activated or reconfigured by software,

one of ordinary skill in the art would also recognize that

such methods may be carried out in hardware, in firmware,

or in more specialized apparatus or devices constructed to

perform the required method steps.

10   A representative computer on which the inventive

operation is performed has a processor (e.g., Intel®-,

PowerPC®- or RISC®-based), random access or other volatile

memory, disc storage, a display having a suitable display

interface, input devices (mouse, keyboard, and the like),

15   and appropriate communications devices for interfacing the

computer to a computer network. Random access memory

supports a computer program that provides the functionality

of the present invention.

Having thus described our invention, what we claim as

20   new and desire to secure by Letters Patent is set forth in

the following claims.